PR4

# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/528,381 | 03/17/2000 | Anindya Basu | Basu 1-1 | 3850 |

7590 05/23/2003

Henry T. Brendzel
P O Box 574
Springfield, NJ 07081

| EXAMINER | |
|---|---|
| ALI, SYED J | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2127 | 3 |

DATE MAILED: 05/23/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PR4

| | | | |
|---|---|---|---|
| **Office Action Summary** | **Application No.**<br>09/528,381 | **Applicant(s)**<br>BASU ET AL. | |
| | **Examiner**<br>Syed J Ali | **Art Unit**<br>2127 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

1)☒ Responsive to communication(s) filed on *17 March 2000* .

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-34* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

## Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)□ The proposed drawing correction filed on _____ is: a)□ approved b)□ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)□ The oath or declaration is objected to by the Examiner.

## Priority under 35 U.S.C. §§ 119 and 120

13)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____ .

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) □ The translation of the foreign language provisional application has been received.

15)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

## Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2 .

4) □ Interview Summary (PTO-413) Paper No(s). _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: .

## DETAILED ACTION

### *Claim Rejections - 35 USC § 112*

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claim 4 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 4, it recites the limitation "said arrangement" in line 21. There is insufficient antecedent basis for this limitation in the claim.

### *Claim Rejections - 35 USC § 103*

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-6, 12-14, and 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bak (USPN 6,212,608) in view of Brandle et al. (USPN 5,218,699) (hereinafter Brandle).

As per claim 1, Bak discloses a system including a processor, and a collection of resources interacting with said processor, said resources including at least a memory and a library of executable modules that are supported by an operating system, the improvement comprising:

a plurality of processing stacks, each including at least one mediation module that processes an applied signal to form a signal that is applied to said at least one resource of said collection of resources (col. 1 lines 37-52, "Each thread has its own execution stack on which method activations reside", wherein each thread of execution operates on system resources through its individual execution stack, and the applied signal is formed out of whatever operations the thread executes).

Bak does not specifically disclose a service director module that intercepts requests of different types that are directed to said resources, classifies said requests in accordance with said types of said requests, and directs said requests to different ones of said processing stacks, based on said classifying.

Brandle discloses a service director module that intercepts requests of different types that are directed to said resources, classifies said requests in accordance with said types of said requests, and directs said requests to different ones of said processing stacks, based on said classifying (col. 3 line 25 – col. 4 line 26, "The service director 12, when called, examines the parameters passed to it and determines which library procedure is to be invoked", wherein the service director handles service requests by determining what resources are attempting to be accessed and then directs the request to the appropriate library).

It is noted that the remote procedure call method of Brandle does not specifically relate to execution stacks. However, Brandle does disclose that depending on the type of system that the invention is implemented on, stacks would be an acceptable way of implementing the disclosed service director (col. 4 lines 27-38, "In other cases, an implementation of one language may pass parameters and return results in registers, while an implementation of another language could use a stack"). It would have been obvious to one of ordinary skill in the art to combine Bak with Brandle since the combination thereof provides a way of ensuring that specific service requests are handled exclusively by the processing stacks associated with a particular resource. Furthermore, by allowing each thread its own execution stack, resource allocation can be monitored and modularized such that memory leaks and security loopholes can be minimized.

As per claim 2, Bak discloses the system of claim 1 wherein said at least one resource to which said signal is applied develops an output signal that is accepted by said at least one mediation module (col. 7 lines 1-60, "Prior to the invocation of method foo 304, header value 314 is stored onto stack 302, as shown in FIG. 3b", wherein the resource to which the signal is applied in this case is a synchronization object, and it develops an output signal including an object header value that tells the process executing that an object is locked).

As per claim 3, Bak discloses the system of claim 1, wherein at least one processing stack of said plurality of processing stacks comprises an ordered sequence of at least two mediation modules (col. 7 lines 1-60, "method foo 304 may invoke other methods. In particular, method foo 304 may transitively invoke another synchronized method", wherein based upon the definition of the word mediation, a mediation module may be any type of software module that acts as a liaison between a thread of execution and system resources. Therefore, if the method foo of Bak may either transitively or the stack successively calls the method bar, the stack can be considered to have an ordered sequence of mediation modules).

As per claim 4, Brandle discloses the system of claim 1, wherein said service director receives a request from an application that is active on said arrangement and applies said request to said at least one mediation module (col. 3 line 25 – col. 4 lines 26, "Once the service director 12 has identified the library procedure to be invoked, it arranges any parameters to be passed thereto appropriately for the calling convention expected by the procedure, and then calls it", wherein the service director passes the function call to the appropriate stack based on the parameters it contains).

As per claim 5, Brandle discloses the system of claim 4, wherein said mediation module receives a return signal from said at least one resource of said collection of resources, processes said return signal to form a processed return signal, and sends said processed return signal to said application (col. 3 line 25 – col. 4 line 26, "Any results returned by the procedure are passed to the service director 12. If necessary, the service director 12 reformats the results, and returns them to the stub procedure").

As per claim 6, Brandle discloses the system of claim 5 wherein said at least one resource of said collection of resources sends said processed return signal via said service director (col. 3 line 25 – col. 4 line 26, "Any results returned by the procedure are passed to the service director 12. If necessary, the service director 12 reformats the results, and returns them to the stub procedure").

As per claim 12, Brandle discloses the system of claim 1, wherein said service director includes:

a service request classifier that classifies a received service request (col. 3 line 25 – col. 4 line 26, "The service director 12, when called, examines the parameters passed to it and determines which library procedure is to be invoked"); and

a processing stack selector that selects a processing stack based upon said classification, and communicates said service request to said selected processing stack (col. 3 line 25 – col. 4 line 26, "Once the desired procedure is identified, the service director must determine where the procedure is stored in the system", "Once the service director 12 has identified the library procedure to be invoked, it arranges any parameters to be passed thereto appropriately for the calling convention expected by the procedure, and then calls it").

As per claim 13, Brandle discloses the system of claim 1, wherein said service director includes a service request classifier that classifies a service request based upon the type of service request and arguments of the service request (col. 3 line 25 – col. 4

line 26, "The service director 12, when called, examines the parameters passed to it and determines which library procedure is to be invoked").

As per claim 14, Brandle discloses the system of claim 1 further comprising a connection to a network (col. 7 lines 21-40, "The remote router 106 communicates with a network interface 110 in order to transfer data and results over the network").

As per claim 24, the modified Bak does not specifically disclose the system of claim 1 wherein said operating system includes means to prevent implication of an operating system breach from an administrative user breach.

"Official Notice" is taken that if the combination of claim 1 was to be implemented, thus only allowing access to the processing stacks through the service director, administrative user breaches would be inherently prevented. That is, since all service requests go through the director, and applications may run within the sandbox without having access to change system resources, modification of the operating system would be impossible in spite of an administrative user breach.

As per claim 25, Brandle discloses the system of claim 1 wherein said service director and said processing stacks are embedded in a loadable library of C language executable modules (col. 3 line 25 – col. 4 line 26, "The application program 16 can perform any end use or system function, and is typically written in a high level language such as C").

3. Claims 7-8, 10-11, 15, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bak in view of Brandle, as applied to claim 1 above, and further in view of Hershey et al. (USPN 5,414,833) (hereinafter Hershey).

As per claim 7, the modified Bak does not specifically disclose the system of claim 1, wherein said at least one mediation module is based upon a chosen security policy.

Hershey discloses the system of claim 1, wherein said at least one mediation module is based upon a chosen security policy (col. 23 lines 33-65, "In this security application, it is assumed that encryption is performed at the application layer", wherein it is discussed above that an application or software module could be construed as a mediation module).

It would have been obvious to one of ordinary skill in the art to combine the modified Bak with Hershey since many security breaches involve corruption of the processing stack. Some examples of known means of breaching the stack include stack override, buffer overflow, smashing the stack, trashing the stack, mangling the stack and others. By implementing a security policy within the processing stack that can handle these types of breaches, a system can be ensured to be more secure.

As per claim 8, the modified Bak does not specifically disclose the system of claim 1, wherein said at least one mediation module in said processing stack performs encryption.

Hershey discloses the system of claim 1, wherein said at least one mediation module in said processing stack performs encryption (col. 23 lines 33-65, "In this security application, it is assumed that encryption is performed at the application layer", wherein it is discussed above that an application or software module could be construed as a mediation module).

As per claim 10, the modified Bak does not specifically disclose the system of claim 1, wherein said mediation module performs authentication.

Hershey discloses the system of claim 1, wherein said mediation module performs authentication (col. 17 lines 8-56, "Upon detecting that program latch 302 has been set, …processor 305 constructs a security alert message from information stored in non-volatile registers 303 and causes a message authentication code to be calculated on the security alert message by invoking data encryption algorithm 304").

As per claim 11, the modified Bak does not specifically disclose the system of claim 1 wherein said mediation module is a secure file system.

"Official Notice" is taken that the implementation of secure file systems is well known in the art. As discussed regarding claim 3, any software module could be treated as a mediation module. Therefore, since the implementation of a secure file system is well known in the art, it would have been obvious to one of ordinary skill in the art to include such a software module in the combination of Bak and Brandle for the purpose of designing a system that is resistant to security breaches. Furthermore, although Hershey does not specifically discuss secure file systems, Hershey does discuss network security, which could be expanded to cover a secure file system without a significant burden.

As per claim 15, the modified Bak does not specifically disclose the system of claim 14 wherein said connection is secure.

As discussed in reference to claim 11, secure file systems would have been an obvious change to the modified Bak based on what is well known in the art. Therefore, if the system therein is said to include a secure file system, it follows that the system would provide a secure connection.

As per claim 23, Hershey discloses the system of claim 1, wherein said at least one mediation module includes at least one authentication code retriever that retrieves an authentication code and a validation system that validates said service request against said authentication code (Security alert message transmission means 306 causes the security alert message and message authentication code to be transmitted via bit stream 124 to a destination device such as a network security manager device", wherein the network security manager device would use the encryption algorithm to validate the authentication code).

4. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bak in view of Brandle as applied to claim 1 above, and further in view of Traversat et al. (USPN 6,052,720) (hereinafter Traversat).

As per claim 9, the modified Bak does not specifically disclose the system of claim 1, wherein said mediation module is a namespace manager.

Traversat discloses the system of claim 1, wherein said mediation module is a namespace manager (col. 7 lines 31-36, "The namespace manager controls how the entries are stored and accessed within the namespace. The manager implements a standard interface that exports the security, storage, and ownership attributes of any entry in the namespace").

It would have been obvious to one of ordinary skill in the art to combine the modified Bak with Traversat since Traversat provides a way of easily routing service requests to the appropriate processing stack. For instance, each particular processing stack may be responsible for performing certain types of tasks. The parameters passed by the application to the service director disclosed by Brandle could contain the destination namespace, and routing of the request to the appropriate stack would be as simple as matching the namespaces.

5. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bak in view of Brandle as applied to claims 1 and 14 above, and further in view of Pierce, Jr. et al. (USPN 6,560,217) (hereinafter Pierce).

As per claim 16, the modified Bak does not specifically disclose the system of claim 14, wherein said network is a virtual private network.

Pierce discloses the system of claim 14, wherein said network is a virtual private network (col. 8 lines 18-39, "Each home agent is associates with one of the virtual private networks, and each home agent has or is associated with a unique IP address").

It would have been obvious to one of ordinary skill in the art to combine the modified Bak with Pierce since it would provide a way of allowing the network interface disclosed by Bak and Brandle in claims 1 and 14 to be implemented in such a way as to provide both the scalability and functionality of a public while also allowing the user the management tools associated with a private network, such as supporting data routing for each processing stack individually.

6. Claims 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bak in view of Brandle in view of Pierce as applied to claims 1, 14, and 16 above and further in view of Hershey.

As per claim 17, the modified Bak does not specifically disclose the system of claim 16 wherein said connection is secured.

Hershey discloses the system of claim 16 wherein said connection is secured. This is discussed above in reference to claims 11 and 15. The discussion of those claims also discusses the added benefit that could be gained by providing a secure network connection, such as preventing security breaches through stack corruption.

As per claim 18, Hershey discloses the system of claim 17 wherein said connection is secured through encryption (col. 23 lines 33-65, "In this security application, it is assumed that encryption is performed at the application layer").

7. Claims 19-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bak in view of Brandle as applied to claim 1 above, and further in view of Bond et al. (USPN 6,275,938) (hereinafter Bond).

As per claim 19, the modified Bak does not specifically disclose the system of claim 1 further comprising a compliance supervisor that is coupled to said processing stacks, and to said service director, and is adapted for receiving security policy information from outside said system.

Bond discloses the system of claim 1 further comprising a compliance supervisor that is coupled to said processing stacks, and to said service director, and is adapted for receiving security policy information from outside said system (col. 8 lines 12-27, "The location of WHKRNL32 352 outside sandbox 215 is especially important, because it is here that the security policy is actually implemented", wherein Bond discloses a module located outside a sandbox that implements a security policy for that sandbox).

It would have been obvious to one of ordinary skill in the art to combine the modified Bak with Bond since it provides an added security benefit to implement the security policy outside of the processing stack. This would prevent an application from modifying the security policy if it were to gain access to the stack, thus cutting off an avenue of attack.

As per claim 20, Bond discloses the system of claim 19, wherein said compliance supervisor receives said security policy information from a virtual private network(col. 1 lines 24-34, "The platform-independent tokenized byte code runs on a virtual machine which places strict limits on what the executable code can do", wherein Bond discloses prior art that shows that to implement particular security policies, particularly in conjunction with sandboxes, is well known to also implement those policies on virtual systems).

As per claim 21, Bond discloses the system of claim 19, wherein said compliance supervisor includes a processing stack modifier that modifies said processing stack based upon a received security policy (col. 7 lines 51-64, "Wx86VM loads API thunk DLLs (secure APIs) such as 391 into the sandbox", wherein the virtual machine loads the security policy from WHKRNL32 into the sandbox, thereby modifying the processing stack in such a way as to adhere to that policy in handling function and memory calls).

As per claim 22, Bak discloses the system of claim 19, wherein said compliance supervisor includes a processing stack creator that creates a processing stack based upon said security policy (col. 1 lines 37-52, "Each thread has its own execution stack on which method activations reside", wherein the creation of a new thread would lead to the creation of a new processing stack. Furthermore, since the virtual machine is controlled by the security policy, whatever processing stacks are created must adhere to that policy).

8. Claims 26-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bak in view of Brandle as applied to claim 1 above, and further in view of Eliott (USPN 6,468,160).

As per claim 26, the modified Bak does not specifically disclose the system of claim 1 further comprising a read-only program store that is read by said system upon boot-up.

Eliott discloses the system of claim 1 further comprising a read-only program store that is read by said system upon boot-up (col. 10 lines 26-56, "serial peripheral interface 138 also includes a 'boot TOM (read only memory)' that stores a small amount of initial program load (IPL) code").

It would have been obvious to one of ordinary skill in the art to combine the modified Bak with Eliott for the purpose of ensuring that the operating system could not be altered during boot-up. By making the program read-only, the operating system could not be breached, and thus the system would become more secure.

As per claim 27, Eliott discloses the system of claim 26, wherein said system includes an operating system, and said read-only program store contains a program module for verifying the operating system, and authentication program modules for authenticating software present in said memory of said system (col. 26 lines 10-16, "The operating system of the video game system 50 is likewise authenticated so that the presence of authentic code in both the video game system and expansion device is verified").

As per claim 28, Eliott discloses the system of claim 27 where said software that is authenticated by said authentication program modules includes software that forms an operating system of said system (col. 26 lines 10-16, "Resident in boot ROM 182 is a set of instructions which permit the remainder of the expansion device operating system to be accessed", wherein the combination of the boot ROM and the expansion device boot system form the operating system for the video game system).

As per claim 29, Eliott discloses the system of claim 28 where said authentication program modules develop a cryptographic hash of software to be authenticated (col. 26 lines 17-21, "Any of various available encryption algorithms may be utilized in order to obtain the desired degree of security", wherein all software on the system is to be authentication using whatever encryption algorithm is chosen to be used in the system).

As per claim 30, it is rejected for similar reasons as stated above for claims 26-29. Specifically, the process of booting up a system and verifying the operating system is shown to be disclosed by Eliott. The additional limitation stating that control is transferred to the operating system once it is verified is well known. That is, once an operating system has been loaded upon a system, control shifts from the kernel to the operating system.

As per claim 31, it is rejected for similar reasons as stated above for claim 29.

As per claim 32, it is rejected for similar reasons as stated above for claim 1. Specifically, the combination of a plurality of processing stacks and a service director make up the reverse sandbox. Therefore, the combination discussed in claim 1 meets this limitation. Furthermore, the discussion of claim 29 shows that all software being loaded on the system of Eliott must be authenticated using whatever encryption algorithm is implemented on the system. To that end, the reverse sandbox would have to be verified against said algorithm before control could be transferred to it.

As per claims 33 and 34, they are rejected for similar reasons as stated above. Any software module, such as a reverse sandbox, would have to be installed on a system in order for it to function in conjunction with the operating system. Furthermore, the service director, compliance modules, processing stacks, and mediation modules are all discussed above thoroughly.

### *Conclusion*

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 1st Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John A Follansbee can be reached on (703) 305-8498. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Syed Ali
May 13, 2003

MAJID BANANKHAH
PRIMARY EXAMINER